New Neoclassical Synthesis' models Numerical Solution

João Pedro Pereira do Couto Rocha

Master's Thesis in Economics

Faculdade de Economia, Universidade do Porto

September 2014

Advised by:
Paulo José Abreu Beleza de Vasconcelos
Pedro Rui Mazeda Gil

Biographic note

João Rocha was born in Vila Nova de Gaia in 1989. Having concluded in 2012 the Licenciate's degree in Management from the Faculty of Economics of the University of Porto, he enrolled, in the same year, in the Master in Economics from the same institution.

Abstract

New Neoclassical Synthesis models, also called Dynamic Stochastic General Equilibrium (DSGE) models, as described by Goodfriend and King (1997), Woodford (2003), or Clarida et al. (1999), emerged in the 1990s as a result of the combination of elements from the Real Business Cycles (RBC) school and New Keynesian school. This synthesis incorporated the prominent role of expectations from the New Classical school, the intertemporal optimising dynamic framework of RBC theory, but also distortions arising from New Keynesian real and nominal rigidities. By being based on microeconomic foundations, and as such seen as less subject to Lucas' critique, these models are useful for, and increasingly used in policy analysis and in economic forecasting tools. Their analytic solution is, however, only possible in certain strict and simpler cases, and as such it is common to resort to approximations instead.

This present work aims to contribute to the literature of New Neoclassical Synthesis and DSGE modelling, focusing on numerical solution methods, which allow approximate solutions to more complex models, namely by applying one of the referred methods, the projection method, to a DSGE model without and with staggered prices.

JEL codes: E30 (Prices, Business Fluctuations, and Cycles), F41 (Open Economy Macroeconomics), C63 (Computational Techniques, Simulation Modelling), C69 (Mathematical methods).

Key words: New neoclassical synthesis; open economy; DSGE; numerical methods; simulation; projection method;

Contents

1 Introduction

New Neoclassical Synthesis models, as described by Goodfriend and King (1997), Woodford (2003), or Clarida et al. (1999), emerged in the 1990s as a result of the combination of elements from the Real Business Cycles (RBC) school and New Keynesian school.

From the RBC school, NNS models inherited its dynamic and stochastic framework, which on its turn was based on the standard neoclassical growth model. Using that framework, RBC models featured economic cycles which arose both from microeconomic decisions of intertemporal profit or utility maximisation from representative agents, and from stochastic behaviour of real variables such as productivity. This intertemporal maximisation depended not only on the expected result of agent's own actions, but also on the expected behaviour of other agents, namely fiscal or monetary authorities modelled in the form of policy rules.

On top of this mathematical framework, some authors began incorporating Keynesian and New Keynesian features, including nominal rigidities such as imperfect competition, or real rigidities such as staggered or sticky price and wage setting. These additions were seen as necessary to improve how these models reflected real data, and to restore a degree of effectiveness to monetary policy, seen by the RBC school as having little or no effect.

In the context of the New Classical school, Lucas (1976) argued against models estimated from historical statistical data as tools for forecasting the effect of economic policies. To support this criticism, Lucas cited the lack of consideration, in the referred practice, for the role of the change of agents' expectations when faced with a given policy, in estimating the outcome of that same policy. In this context, RBC models' microeconomic foundations were seen as making them less subject to Luca's critique. By incorporating the expected actions of authorities into the way how representative agents acted, any possible change in the former's behaviour would already be reflected in the later's current decisions.

This perceived robustness was inherited by NNS models, which had incorporated the referred Keynesian rigidities as deriving, too, from microeconomic behaviour of households and firms. This, together with the added realism from the referred rigidities, turned them into popular tools of policy analysis and economic forecasting.

Dynamic Stochastic General Equilibrium models, which include the NNS, however, don't always have analytic solutions, except in the case of smaller models or in the presence of certain function specifications. Nevertheless, numerical methods, in this particular case, but also in economics generally, allow obtaining approximate answers to large dimensional and more complex problems. Most of the literature of this school, then, relies on numerical approximation techniques.

The solution to Dynamic General Equilibrium models is usually defined either by a set of Euler equations, resulting from the application of the Lagrange multipliers method, or a policy function, resulting from the use of dynamic programming. Numerical solution methods can be categorised by the kind of solution definition they apply to, that is, the type of function they attempt to approximate: Euler equations or policy function. A second categorisation of methods can be made based on whether the approximation is local (only valid near a point) or global (valid in the whole domain).

One of the most widely used approximations is the family of perturbation methods, which rely on Taylor series expansions of different orders of the problem's Euler equations around a point, usually its steady-state. Often a single Taylor expansion is used, as in the case of linearisation or log-linearisation – two particular applications of this family of methods – which may exclude important features of the original problems. For this reason, second and higher order have been proposed and are applied. These techniques are faster than most, but their locality precludes the analysis of larger shocks that imply bigger deviations from the point of approximation.

Discrete State Space methods, on the other hand, are global methods that can be applied to policy function problems. They involve approximating the referred function through a recursive iterative process, called policy function iteration, after imposing a fixed grid of possible values for its variables. Although global and always converging to the true function with enough iterations, this method can become too computationally intensive with an increase of the complexity of the problem, or when finer grids are employed.

This work applies a third solution option, consisting of the Projection method, a global method with similarities to regression that applies to functional problems, such as those defined by either policy function or Euler equations (Judd, 1998). This method starts with the choice of a generic approximation function, such as Chebychev polynomials or Artificial Neural Networks, which is meant to replace the unknown of the functional problem. Then, a measure of distance between the approximated and the true function, or a residual function, is found. Finally, a method of parameterising the generic function in order to minimise the referred residual is chosen, such as the Least squares method, Chebychev, Galerkin or other collocation method. Being a global method, the projection method doesn't preclude the analysis of larger shocks. Additionally, depending on the used approximation function, the full nonlinearities of the problem can be preserved. However, the larger the model or the number of approximation function parameters to be calculated, the higher its computational cost.

We start, then, with a model based on Lim and McNelis (2008), similar to the standard

NNS or New Keyensian model, which is subsequently extended with staggered adjustment of prices. At each step the projection method is used, and the necessary changes to the solution method in the different model versions explained. Finally, a set of simulations and impulse-response shocks are used to showcase the model, extension, method and application.

In the next chapter, a literature review on the New Neoclassical Synthesis and DSGE models is presented, and numerical solution methods explored, not neglecting some analytic approaches.

2 Literature review

2.1 New Neoclassical Synthesis

The original Neoclassical Synthesis, first described by Paul Samuelson, as cited by Blanchard (2008), and typified in models such as IS/LM (Hicks, 1937), conciliated the Keynesian and New Classical views, with sticky prices in the short term causing imbalances that could be fought with the adequate monetary and fiscal policies, and flexible prices in the long term.

This synthesis, with the arrival of the rational expectations in the 70s, was followed by the New Classical school. Assuming fully flexible prices, and the idea that agents utilise in a perfectly rational way all available information, reflecting it in their expectations, this last school criticised the former and emphasised, instead, the role of imperfect availability of information on the departure of economies from equilibrium. This assumption of rational expectations came as a response to Lucas (1976), who argued against the use of models with parameters estimated from historical data as forecasting instruments. This criticism rested on the observation of the fact that those models did not take into account the agents' expectations and behaviour changes, when faced with policy changes, while testing the effect of those same policies.

The New Keynesian school, emerged in the 80s, tried, on its turn, to simultaneously accept and incorporate New Classical school critiques to the original neoclassical synthesis, namely Lucas' critique and rational expectations. At the same time, it sought to reintroduce Keynesian elements, namely sticky prices and wages, perceived as more realist assumptions. This was achieved, at first, by the inclusion of real rigidities, such as imperfect competition. While not a source of price stickiness by itself, by providing agents a measure of market power, the imperfect competition framework turned them into price-setters, deciding based on costs and profits, contrasting with the price-takers of the New Classical school. This addition, on the other hand, allowed the introduction of nominal rigidities in price adjustment mechanisms, such as frictions in price and wage setting requiring agents to set prices only in certain conditions (Goodfriend and King, 1997).

The reintroduced imperfect price adjustment was modelled initially as being sourced in small menu costs (costs associated with changes and announcements of new prices)(Akerlof and Yellen, 1985; Mankiw, 1985). Later, in the context of dynamic models, wage adjustment mechanisms were introduced, such as that proposed by Taylor (1979), where different groups of workers saw their wage updated in given fixed intervals and distinctly for each group. Additionally, staggered price adjustment mechanisms were introduced, such as that proposed by Calvo (1983), which made prices fixed for limited but random periods of time.

Parallel to those developments, and framed within the context of the New Classical school, the Real Business Cycle (RBC) school and models arose, with the initial contribution of Kydland and Prescott (1982). In that work, its authors repurposed one of the standard models of neoclassical growth theory, the Ramsey-Cass-Koopmans model (Ramsey, 1928; Cass, 1965; Koopmans, 1965), for the study of economic cycles. Inheriting that model's structure, RBC models derived the behaviour of the economy from the behaviour of utility maximising representative agents. Subject to the available technology, budget constraints, and their own preferences, those same agents chose between consumption or leisure and investment, in a dynamic framework in which each decision affected current and future payoffs. Since the referred maximisation involved not only the current but also all discounted future utility, agents acted in a forward-looking manner. This involved taking into account both their action's consequences and expected actions of government agents, modelled in the form of policy rules, responding in this manner to Lucas' critique (McCandless, 2008).

In respect to the sources of economic cycles, the conclusions of Kydland and Prescott in the cited article, and in most of the RBC literature, as surveyed by Rebelo (2005), emphasised the role of real effects such as stochastic technology shocks to productivity, amplified by other features such as building time of investments or changes in inventory stocks. The subsequent addition of money or monetary policy to those models, on the other hand, was found to have little or no effect on generating fluctuations. Complications related to the fit of these models to the data, which required parameters and shocks of magnitudes on its turn not found in the data, led to different additions and modifications of the base RBC framework, including energy prices, fiscal or terms of trade shocks.

In this context, by the end of the 90s, authors like Goodfriend and King (1997), Woodford (2003), or Clarida et al. (1999), began describing a new approach, to which they called New Neoclassical Synthesis (NNS) or New Keynesian Synthesis. These new models based themselves on RBC's optimising representative agents structure, on top of which Keynesian and New Keynesian elements were introduced. Contrasting with RBC models, where its representative agents were price takers in perfectly competitive markets, NNS models incorporated imperfect competition and price rigidities from the New Keynesian school. Unlike the original neoclassical synthesis, where price behaviour was linked to measures of market disequilibrium, here it arose from the representative agent's microeconomic behaviour, becoming less fit for the Lucas' critique (Woodford, 2003). At the same time, staggered prices and imperfect competition allowed a degree of effectiveness of monetary policy in affecting the real economy, contrasting once again with the RBC school. This combination turned these models into popular tools for answering questions related to monetary policy and central banking[1], or fiscal policy[2], for example. Furthermore, some authors started using the NNS model as a basis for larger models, containing a large and diverse number of shocks and frictions, estimated for existing economies, and with the aim of doing policy analysis, such as Smets and Wouters (2003).

The model whose numerical solution this work intends to study, described in Lim and McNelis (2008), can be included on this last school, and will be described further ahead.

2.2 Numerical solution methods

The solution to the underlying optimisation problems of NNS, and Dynamic General Equilibrium models in general, is usually obtained via dynamic programming (Stokey and Prescott, 1989) or via application of the Kuhn-Tucker theorem or Lagrange multipliers method (Chow, 1997). Both approaches turn the optimisation problem into a functional one. In the case of dynamic programming, its solution is the policy function, which returns the optimising choice of control variables, for each given value of the state variables. Similarly, the Lagrange multipliers approach results in one or more Euler equations, forming a system of difference equations (differential equations, for continuous time) which describe the optimal path taken by the control variables that solves the optimisation problem.

Regardless of the solution approach, analytic solutions for these models only exist in certain restrictive conditions, such as certain utility or production function specifications, described in Canova (2007) or Heer and Mauß ner (2009), for example. Given the impossibility or, in certain cases, the difficulty of obtaining such analytic solutions, it is common to resort to approximation techniques instead.

Heer and Mauß ner (2009) categorise approximation techniques using two types of features. The first one is related to the function it tries to approximate: policy function or Euler equation. The second one is concerned with the kind of approximation, which can be local or global. Local techniques try to obtain a function that only approximates the true one around a given point, usually its steady-state, contrasting with global ones, which usually aim to obtain valid approximations for the whole domain.

Linear approximation is a common local method, which involves turning the original problem's first order conditions or constraints into approximate linear substitutes. Different linearisation techniques exist, including Log-linear and Linear quadratic (LQ) approxima-

[1] The effectiveness of different monetary rules, for example (Clarida et al., 1999)

[2] Through the addition or improvement of the base NNS model's fiscal section (Galí et al., 2007)

tion. Log-linearisation applies to both the Euler equation and budget constraints, which are approximated by taking logarithms and first-order Taylor series expansions around the problem's steady-state, subsequently solving the resulting linearised system. Uhlig (1995) is an example of this approach. LQ approximation, on the other hand, consists of quadratic approximations for the problem's objective function and linear approximations for its constraints, once again around its steady-state.

These linear approximation methods are included in the larger group of perturbation methods, which also resort to Taylor expansions of different orders around the problem's steady-state. They rest on the assumption that the modelled economy never diverges too far away from its steady-state, so that higher order members of the Taylor series can be ignored without severe accuracy loss. These methods are less computationally intensive than others, and in most cases the assumption is acceptable, and the approximation good enough. As noted by Kim and Kim (2003), however, although a wide range of the literature involving DSGE models relies on linear approximations for their solution, their use may result in high approximation errors and distort the results of their analysis. For this reason, several second and higher-order perturbation methods and algorithms have been advanced (Swanson et al., 2005; Schmitt-Grohé and Uribe, 2004).

Another kind of techniques, Discrete State Space methods, also called Discretisation, are global methods that can be applied to policy function problems. According to Canova (2007), they generally involve forcing the states and exogenous variables to take values from a fixed grid of possible values. This, on the other hand, facilitates obtaining the policy function through a recursive iterative process called policy function iteration. This approach always converges to the true policy function. However, with an increase of the number of variables or shocks, the problem may become too highly computationally intensive, in what is known as the curse of dimensionality. Furthermore, this approach involves a trade-off between speed and accuracy: the finer the grid, the better the approximation, but the higher the number of computations involved.

A third alternative, the one applied in this work, are Projection methods. Described in Judd (1998), these are global methods that apply to problems with solution defined either by policy function or Euler equations. They involve choosing an approximation function for each of the decision rules, which in turn is tuned within an iterative process controlled via a chosen residual function. This way, the functional problem is turned into one of minimising the residual function. Variations of the method are possible given the different existing choices of approximating function (Chebychev polynomials or neural networks, for example), residual function or minimisation procedure (Least squares, Galerkin method or other collocation methods, for example). This approach, like other global methods, does not rely on the smallness of shocks for a good approximation. Additionally, the validity of the approximation it not restricted to the neighbourhood of the point around which it is made. However, the better approximation comes with a greater computational cost, especially when a large number of parameters has to be calculated, such as in the case of higher order polynomials or more complex neural networks.

3 Models' exposition and numerical solution

As previously mentioned, the particular model to be studied in the present work is based on Lim and McNelis (2008). It consists of a dynamic in discrete time, stochastic general equilibrium model.

It is constituted by three main sectors in an open economy: households, firms and monetary authorities; and two smaller ones, namely fiscal authorities and rest of the world. The household sector decides at each time period between consumption and leisure, subject to a budget constraint. This budget constraint includes, among other items, the firm's profits, given that the former are fully owned by households. The price dynamics of the model stem from production and pricing decisions of the domestic (consumption and intermediate) goods producing firms, and the monetary authority acts on inflation by setting the interest rate via a simple Taylor rule. On its turn, the fiscal authority is presented as an exogenously defined amount of government spending and lump-sum taxes. Finally, capital or producer goods, which fully depreciate at the end of each time period, are imported by households from the rest of the world, being subsequently lent to domestic firms.

This section presents the model and its numerical solution in two parts. Firstly, a baseline version of the model is presented, with fully flexible prices. Secondly, this base model is modified with the addition of sticky or staggered prices. Every subsection includes the details of the corresponding numerical solution, namely the modifications required for the referred addition.

3.1 Standard (flexible-prices) version

3.1.1 Households

The household sector's choice between consumption and leisure translates, in practice, to an optimisation problem as follows.

Firstly, the representative household values its present and future consumption according to the function

$$V = E_0 \sum_{t=0}^{\infty} \beta^t U_t \left(C_t, L_t\right) \tag{3.1}$$

where its preferences are represented by U_t, the utility function, which is of the type

$$U_t(C_t, L_t) = \frac{C_t^{1-\eta}}{1-\eta} - \frac{L_t^{1+\varpi}}{1+\varpi} \tag{3.2}$$

Value function equation (3.1) constitutes the expected present value, given the discount factor β, for the representative household, of both current and all future utility. This utility, as seen on equation (3.2), depends positively on the amount of consumption C_t, and negatively on the amount of labour L_t of the corresponding period.

The utility function is of Constant Relative Risk Aversion type. The measure of the risk aversion curve, η, is the relative risk aversion coefficient (or elasticity of marginal utility of consumption), and ϖ the elasticity of marginal disutility of labour.

C_t is an index of consumption goods, given on its turn by

$$C_t = \left[\int_0^1 \left(C_{j,t} \right)^{(\zeta-1)/\zeta} dj \right]^{\zeta/(\zeta-1)}, \tag{3.3}$$

a Dixit-Stiglitz, or Armington, aggregator (Dixit and Stiglitz, 1977). Here, an infinite number of goods, indexed by j, become perfect substitutes when ζ, the constant elasticity of substitution coefficient, approaches infinity, and perfect complements when it approaches zero.

The choice that maximises the utility function above is, on other hand, restricted by the budget constraint

$$W_t L_t + \Pi_t + P_t^K K_t + (1 + R_{t-1}) B_{t-1} + S_t F_t = P_t C_t + P_t^f I_t + B_t + \left(1 + R_{t-1}^* + \Phi_{t-1} \right) S_t F_{t-1} + T_t \tag{3.4}$$

with the variables representing:

- W_t, the wages;
- Π_t, the firms' profits;
- K_t, the stock of capital goods, which fully depreciates at each period, and as such equals I_t, the capital imports;
- P_t^K, the capital goods' price, charged by households to firms, with the former importing the same goods for P_t^f, the imported capital goods' price;
- B_t, the stock of domestic government bonds bought by households, and F_t, the stock of bonds issued by households, or external debt;
- S_t, the exchange rate (expressed in indirect quotation);

- Φ_t, a risk premium;
- and T_t, the amount of lump-sum taxes.

The Lagrangian for the maximisation of equation (3.1) with respect to (3.4) is as follows:

$$\mathscr{L} = E_0 \sum_{t=0}^{\infty} \beta^t \Big\{ U\left(C_{t,} L_t\right) - \lambda_t \Big[P_t C_t + P_t^f K_t + B_t + \left(1 + R_{t-1}^* + \Phi_{t-1}\right) S_t F_{t-1}$$
$$+ T_t - W_t L_t - \Pi_t - P_t^K K_t - (1 + R_{t-1}) B_{t-1} - S_t F_t \Big] \Big\}$$

Obtaining and solving the first order conditions

- $\frac{\partial \mathscr{L}}{\partial C_t} = \frac{\partial U}{\partial C_t} - \lambda_t P_t = 0$
- $\frac{\partial \mathscr{L}}{\partial L_t} = \frac{\partial U}{\partial L_t} + \lambda_t W_t = 0$
- $\frac{\partial \mathscr{L}}{\partial K_t} = \lambda_t P_t^K - \lambda_t P_t^* = 0$
- $\frac{\partial \mathscr{L}}{\partial F_t} = \beta^t \lambda_t S_t - \beta^{t+1} \lambda_{t+1} \left[\Phi' S_{t+1} F_t + (1 + R_t^* + \Phi_t) S_{t+1}\right] = 0$
- $\frac{\partial \mathscr{L}}{\partial B_t} = -\beta^t \lambda_t + \beta^{t+1} \lambda_{t+1} (1 + R_t) = 0 \Leftrightarrow \beta \lambda_{t+1} (1 + R_t) = \lambda_t$

and rearranging, results in the Euler equations of the problem, describing the path that solves the intertemporal maximisation problem:

$$\frac{C_t^{-\eta}}{P_t} = \frac{C_{t+1}^{-\eta}}{P_{t+1}} \beta (1 + R_t) \tag{3.5}$$

$$W_t = L_t^{\varpi} P_t C_t^{\eta} \tag{3.6}$$

$$P_t^* = P_t^K \tag{3.7}$$

$$(1 + R_t) S_t = \left(1 + R_t^* + \Phi_t' F_t + \Phi_t\right) S_{t+1} \tag{3.8}$$

3.1.2 Firms

The production sector is made up of two types of firms: intermediate goods firms, and final goods firms. The former hire labour and capital goods from households, paying wages and the capital rent price in return, combining both into intermediate goods. The later buy the referred intermediate goods, which are combined into a final good subsequently sold to households. The production and pricing choices of both kinds of firms will determine the price dynamics of the model.

Intermediate goods firms Each intermediate goods firm, indexed by $j \in [0;1]$, employs, in each period t, capital goods (K_t) and labour (L_t) as factors of production, producing $Y_{j,t}$ according to the Constant Elasticity of Substitution production function

$$Y_{j,t} = Z_t \left[(1-\alpha)\left(L_{j,t}\right)^{\kappa} + \alpha (K_{j,t})^{\kappa}\right]^{1/\kappa} \tag{3.9}$$

The transformation $\frac{1}{1-\kappa}$ stands for the elasticity of substitution between the two factors of production[3], and α for the usage share of the same factors[4]. Z_t, on the other hand, is a random productivity shock, which follows an autoregressive process around its steady-state value $\overline{Z}$, subject to a random disturbance ε_t^Z as follows:

$$\ln(Z_t) = \rho \ln(Z_{t-1}) + (1-\rho)\ln\left(\overline{Z}\right) + \varepsilon_t^Z,\ \varepsilon \sim N\left(0, \sigma_Z^2\right) \tag{3.10}$$

Intermediate goods firms decide as to maximise profits by setting the produced quantity, given by $Y_{j,t}$. Profit maximisation implies, on its turn, (constrained) minimisation of total costs:

$$\min_{L_{j,t}, K_{j,t}} TC_{j,t} = W_{j,t} L_{j,t} + P_{j,t}^K K_{j,t} \tag{3.11}$$
$$s.t. (3.9)$$

Setting up the Lagrangian,

$$\mathscr{L} = W_{j,t} L_{j,t} + P_{j,t}^K K_{j,t} - \lambda_{j,t}\left(Y_{j,t} - Z_t\left[(1-\alpha)\left(L_{j,t}\right)^{\kappa} + \alpha\left(K_{j,t}\right)^{\kappa}\right]^{1/\kappa}\right)$$

solving and simplifying the first order conditions results into

[3] $0 < \kappa < 1$
[4] $0 < \alpha < 1$

$$\frac{\partial \mathscr{L}}{\partial L_{j,t}} = 0 \Leftrightarrow \frac{W_t}{(1-\alpha)L_{j,t}^{\kappa-1}} = \lambda_t Z_t \left[(1-\alpha)\left(L_{j,t}\right)^{\kappa} + \alpha\left(K_{j,t}\right)^{\kappa}\right]^{1/\kappa - 1}$$
$$\frac{\partial \mathscr{L}}{\partial K_{j,t}} = 0 \Leftrightarrow \frac{P_t^K}{\alpha K_t^{\kappa-1}} = \lambda_t Z_t \left[(1-\alpha)\left(L_{j,t}\right)^{\kappa} + \alpha\left(K_{j,t}\right)^{\kappa}\right]^{1/\kappa - 1}$$

which can be rearranged into the optimal factor combination

$$K_{j,t} = \left(\frac{W_t}{(1-\alpha)} \cdot \frac{\alpha}{P_t^K}\right)^{\frac{1}{1-\kappa}} L_{j,t} \tag{3.12}$$

Combining the above equation with production function (3.9), and solving for L and K, returns the firm's conditional demands for labour and capital, respectfully.

$$\begin{aligned} L_{j,t} &= \left(\frac{Y_{j,t}}{Z_t}\right)\left[(1-\alpha) + \alpha\left(\frac{\alpha W_t}{(1-\alpha)P_t^K}\right)^{\kappa/(1-\kappa)}\right]^{-1/\kappa} \\ K_{j,t} &= \left(\frac{Y_{j,t}}{Z_t}\right)\left[\alpha + (1-\alpha)\left(\frac{(1-\alpha)P_t^K}{\alpha W_t}\right)^{\kappa/(1-\kappa)}\right]^{-1/\kappa} \end{aligned}$$

Finally, the choice of quantity produced can be translated into the optimisation problem:

$$\max_{Y_{j,t}} \Pi_{j,t} = P_{j,t}Y_{j,t} - W_{j,t}L_{j,t} + P_{j,t}^K K_{j,t}$$

Plugging the conditional demands for labour and capital into the corresponding Lagrangian of the problem above

$$\mathscr{L} = P_{j,t}Y_{j,t} - \frac{Y_{j,t}}{Z_t}\left[W_t\left[(1-\alpha) + \alpha\left(\frac{\alpha W_t}{(1-\alpha)P_t^K}\right)^{\kappa/(1-\kappa)}\right]^{-1/\kappa} + P_t^K\left[\alpha + (1-\alpha)\left(\frac{(1-\alpha)P_t^K}{\alpha W_t}\right)^{\kappa/(1-\kappa)}\right]^{-1/\kappa}\right]$$

and solving the first order condition, returns the profit maximisation condition:

$$P_{j,t}=\frac{1}{Z_t}\left[W_t\left[(1-\alpha)+\alpha\left(\frac{\alpha W_t}{(1-\alpha)P_t^K}\right)^{\kappa/(1-\kappa)}\right]^{-1/\kappa}+P_t^K\left[\alpha+(1-\alpha)\left(\frac{(1-\alpha)P_t^K}{\alpha W_t}\right)^{\kappa/(1-\kappa)}\right]^{-1/\kappa}\right] \tag{3.13}$$

The right side of equation (3.13) is the marginal cost of intermediate goods firms[5]. By using $A_{j,t}$ to denote it, the profit maximisation condition can be simplified to:

$$P_{j,t}=A_{j,t}$$

Final goods firms. As described previously, the final goods firms take the intermediate goods as production inputs, transforming them into a single final good, Y_t, according to the aggregating function

$$Y_t=\left[\int_0^1\left(Y_{j,t}\right)^{(\zeta-1)/\zeta}dj\right]^{\zeta/(\zeta-1)} \tag{3.14}$$

Taking both final good P_t and intermediate goods price $P_{j,t}$ as given, it maximises profits:

$$\max_{Y_{j,t}} P_tY_t-\int_0^1 P_{j,t}Y_{j,t}dj \tag{3.15}$$

Setting up and solving the first order condition while taking 3.14 into account

$$\frac{\partial}{\partial Y_{j,t}}\left[P_t\left[\int_0^1\left(Y_{j,t}\right)^{(\zeta-1)/\zeta}dj\right]^{\zeta/(\zeta-1)}-\int_0^1 P_{j,t}Y_{j,t}dj\right]=0$$

leads to (3.16), which can be seen as the demand faced by intermediate goods firms as the final goods firm produces Y_t:

$$Y_{j,t}=\left[\frac{P_{j,t}}{P_t}\right]^{-\zeta}Y_t \tag{3.16}$$

3.1.3 Monetary and fiscal authorities

The monetary authority adjusts the interest rate (R_t) partially every period, according to a Taylor rule that takes into account both inflation (π_t) and external interest rate (R_t^*), with

[5] $\frac{\partial TC_t}{\partial Y_{j,t}}=\frac{1}{Z_t}\left[W_t\left[(1-\alpha)+\alpha\left(\frac{\alpha W_t}{(1-\alpha)P_t^K}\right)^{\kappa/(1-\kappa)}\right]^{-1/\kappa}+P_t^K\left[\alpha+(1-\alpha)\left(\frac{(1-\alpha)P_t^K}{\alpha W_t}\right)^{\kappa/(1-\kappa)}\right]^{-1/\kappa}\right]$

adjustment parameters ϕ_1 and ϕ_2[6]:

$$R_t = \phi_2 R_{t-1} + (1-\phi_2)\left(R_t^* + \phi_1 \pi_t\right) \tag{3.17}$$

Inflation, on its turn, is defined as

$$\pi_t = \left[\left(\frac{P_t}{P_{t-1}}\right)^4 - 1\right] \tag{3.18}$$

As for the fiscal authority, its role in the model is simplified to a single constant exogenous variable for public spending, $G = \overline{G}$.

3.1.4 Rest of the world

The rest of the world comes into the model both as a producer of capital goods, as an importer and as a lender of one period bonds. Capital goods are imported by households and subsequently rented to firms. Exports to the rest of the world, on its turn, are introduced via a constant exogenous variable $X = \overline{X}$. The relationship between the trade balance and the domestic economy's external debt is described by the equation

$$P_t X_t - S_t P_t^K K_t = \left(1 + R_{t-1}^* + \Phi_{t-1}\right) S_t F_{t-1} - S_t F_t \tag{3.19}$$

meaning that any trade balance[7] deficit (superavit) will increase (decrease) the economy's external debt (given by $S_t F_t$).

The interest paid on external debt bonds includes Φ_t, a risk premium, which increases as the external financial position, or external debt, increases above its steady-state $\overline{F}$ (according to sensitivity parameter $\varphi > 0$), and is given by

$$\Phi_t = sign\left(F_t\right) \times \varphi \left[e^{\left(|F_t| - \overline{F}\right)} - 1\right] \tag{3.20}$$

3.1.5 Closing the model

The remaining equation required to complete the model is the aggregate demand, or demand for final, or consumption, goods:

$$Y_t = C_t + G_t + X_t \tag{3.21}$$

[6] $0 < \phi_1 < 1, 0 < \phi_2 < 1$

[7] the left side of (3.19)

The above expression, in addition to equations (3.9), (3.12), (3.13), (3.17) and (3.19), and Euler equations (3.5), (3.6), (3.7) and (3.8) form the full system that describes the model, to be solved for C_t, S_t, R_t, W_t, L_t, P_t^K, K_t, P_t, F_t and Y_t.

3.1.6 Numerical solution

The chosen solution method for the simulation of the model consists of the Projection Method, as described by Judd (1998) and placed into context earlier in this work. It involves:

- Choosing an approximation function specification for each of the ones to be approximated;
- Choosing a measure of the approximation error;
- Choosing an optimisation algorithm to iterate until the chosen error is reduced to a chosen level.

Approximation function. In this case, the functions to be approximated include Euler difference equations (3.5) and (3.8), which describe the optimal path of C_t and S_t.

The approximation function specification consists of an artificial neural network, namely a multilayer perceptron. Each variable is approximated by a single neuron, denoted by n, with three input nodes z – one for each of the state variables Z_t, F_t and R_t, defined as

$$n_i = f\left(\sum_{k=1}^{j} w_k z_k\right)$$

where i is the index of approximated functions, k the index of the neuron's inputs, and with activation function $f(x) = \dfrac{1}{1+e^{-x}}$. Finally, the approximation is parameterised by weights w.

Approximation error. The error measures used will be based on the approximated Euler equations. Rearranging the referred equations, and letting $\hat{C}_t$ and $\hat{S}_t$ denote the approximated values of C_t and S_t respectively, results into the Euler errors

$$\varepsilon_t^c = \frac{\hat{C}_t^{-\eta}}{P_t}\left[\frac{1}{1+R_t}\right] - \beta\left[\frac{\hat{C}_{t+1}^{-\eta}}{P_{t+1}}\right] \tag{3.22}$$

$$\varepsilon_t^S = (1+R_t)\hat{S}_t - \left(1+R_t^* + \Phi_t' F_t + \Phi_t\right)\hat{S}_{t+1} \tag{3.23}$$

which measure the distance between the approximated and the true values – those that verify the optimal path as described by the Euler equations.

Approximation algorithm. The computational implementation of the model is made up of three separate parts. The first one consists of an algorithm, encapsulated in a MATLAB function, described below, which simulates the approximated model and returns the simulation's Euler errors, for any given set of parameters and neural network weights. The second consists of a MATLAB script that iterates on the referred function, based on an initial guess of the approximation weights, until the sum of squared Euler errors are minimised, according to a previously chosen tolerance. Finally, a third script simulates the model as-is, by calling the model's simulation function, and where impulse-response shocks, for example, can be defined.

In all simulations, the initial values of all variables are set to the corresponding steady-state value. This value, on its turn, is calculated with an helper function, detailed in Appendix 6.1.

Algorithm 1 Model simulation function

```
initialise model parameters, steady-state values and auxiliary variables
for i = 1 to t (number of simulation periods)

    draw values for random processes (Z_t)
    get output of neurons (Ĉ_t , Ŝ_t)
    numerically solve rest of system using MATLAB's fsolve, ...
        obtaining Y_t , L_t , K_t , P_t^K , P_t , W_t , R_t , and F_t
    calculate euler errors (ε_t^S , ε_t^C)

end loop
```

3.2 Sticky prices version

A modifications is now introduced, designed to allow for staggered, or sticky, behaviour of prices.

There are several ways by which sticky prices are usually included in NNS models. The one chosen here is a Calvo rule (Calvo, 1983), which in practice separates, for each time period, intermediate goods firms into two groups: one allowed to change prices, and another that is not. Accordingly, firms that can change prices will do so by setting the price that maximises its present and discounted future profits, taking into account both the demand they face and the probability of being able to set prices again in the future. The ability to set prices, on its turn, arises from the assumption of imperfect competition already introduced in the baseline version, caused by imperfect substitutability between intermediate goods firms' differentiated products.

We start by detailing the extension to the baseline version of the model addition by describing the specific changes to the firm behaviour referred to above.

3.2.1 Firms

As in the first version of the model there are two kinds of firms, intermediate and final goods producers.

The final goods firm**,** as before, maximise their profits as given by (3.15), subject to the aggregator production function (3.14). The solution to this problem, on its turn, returns the demand for intermediate goods (3.16).

Intermediate goods firms' intertemporal profit maximisation problem, on the other hand, starts with the implied cost minimisation. As in the flexible price version, plugging the optimal factor combination (3.12) into total cost equation (3.11) and differentiating with respect to quantity produced $Y_{j,t}$ returns the intermediate goods firms' marginal cost, reproduced again here:

$$A_{j,t} = \frac{1}{Z_t}\left[W_t\left[(1-\alpha)+\alpha\left(\frac{\alpha W_t}{(1-\alpha)P_t^K}\right)^{\kappa/(1-\kappa)}\right]^{-1/\kappa} + P_t^K\left[\alpha+(1-\alpha)\left(\frac{(1-\alpha)P_t^K}{\alpha W_t}\right)^{\kappa/(1-\kappa)}\right]^{-1/\kappa}\right]$$

Amongst intermediate goods firms, the ones that can change prices do so, as previously mentioned, by maximising the present value of their profits. With θ being the probability of a given intermediate goods firm changing its price on a given period, $P_{j,t}^A$ its chosen price, and μ a subsidy which will be explained further ahead, the present value of the firm's profits is

$$V = E_t\sum_{t=0}^{\infty}\theta^t\beta^t\left((1+\mu)P_{j,t}^A Y_{j,t} - A_{j,t}Y_{j,t}\right)$$

which by combining with intermediate good's demand (3.16) becomes

$$V = E_t\sum_{t=0}^{\infty}\theta^t\beta^t\left((1+\mu)P_{j,t}^A\left(\frac{P_{j,t}^A}{P_t}\right)^{-\zeta} - A_{j,t}\left(\frac{P_{j,t}^A}{P_t}\right)^{-\zeta}\right)Y_t$$

Maximising the present value of the firm's profits above with respect to price P^A returns the expression for the optimal price:

$$
\begin{aligned}
\frac{\partial}{\partial P_{j,t}^A}\left[E_t\sum_{t=0}^{\infty}\theta^t\beta^t\left((1+\mu)P_{j,t}^A\left(\frac{P_{j,t}^A}{P_t}\right)^{-\zeta}-A_t\left(\frac{P_{j,t}^A}{P_t}\right)^{-\zeta}\right)Y_t\right]=0 \\
\Leftrightarrow E_t\sum_{t=0}^{\infty}\theta^t\beta^t\left[(1+\mu)\left[\frac{\left(P_{j,t}^A\right)^{-\zeta}}{P_t^{-\zeta}}-\zeta\left(P_{j,t}^A\right)^{-\zeta-1}(P_t)^{\zeta}P_{j,t}^A\right]+A_t\zeta\left(P_{j,t}^A\right)^{-\zeta-1}P_t^{\zeta}\right]Y_t=0 \\
\Leftrightarrow \frac{\zeta}{\zeta-1}\cdot\frac{1}{1+\mu}\cdot\frac{E_t\sum_{t=0}^{\infty}\theta^t\beta^t A_t(P_t)^{\zeta}Y_t}{E_t\sum_{t=0}^{\infty}\theta^t\beta^t(P_t)^{\zeta}Y_t}=P_{j,t}^A
\end{aligned}
\tag{3.24}
$$

Here, the previously referred subsidy μ is chosen in order to cancel the effect of $\frac{\zeta}{\zeta-1}$, which would otherwise act as a markup over the competitive price, caused by the market power of intermediate goods firms. This degree of market power is, on its turn, given by elasticity of substitution in the aggregator production function (3.14).

Having derived the price set by firms that are allowed to do so, one needs to combine it with the price of the remaining firms to obtain the aggregate price index, which is done using a Dixit-Stiglitz aggregator as follows:

$$
P_t=\left[\theta\left(P_{t-1}\right)^{1-\zeta}+(1-\theta)\left(P_t^A\right)^{1-\zeta}\right]^{1/(1-\zeta)}
\tag{3.25}
$$

Apart from the firms' section, the rest of the model carries on unmodified to the sticky price version.

3.2.2 Numerical solution

Solving the model, in comparison to its original version, requires approximating the additional difference equation (3.25). Both the numerator and denominator of the latter equation can be rewritten as a recursive relationship by using two auxiliar variables, A_t^{p1} and A_t^{p2}, defined as

$$
\begin{aligned}
A_t^{p1} &= A_t(P_t)^{\zeta}Y_t+\theta\beta A_{t+1}^{p1} \\
A_t^{p2} &= (P_t)^{\zeta}Y_t+\theta\beta A_{t+1}^{p2}
\end{aligned}
$$

This, in turn, allows (3.24) to be rewritten as

$$P_t^A = \frac{A_t^{p1}}{A_t^{p2}} \tag{3.26}$$

For the numerical approximation, two additional neurons will be employed, one for each of the auxiliar variables above.

Approximation error Using $\hat{A}_t^{p1}$ and $\hat{A}_t^{p2}$ as the approximate versions of A_t^{p1} and A_t^{p2} respectively, and optimal price recursive equation (3.26), a new Euler error can be written:

$$\varepsilon_A = \frac{\hat{A}_t^{p1}}{\hat{A}_t^{p2}} - \frac{A_t \left(P_t\right)^{\zeta} Y_t + \theta \beta \hat{A}_{t+1}^{p1}}{\hat{A}_t^{p2} - \left(P_t\right)^{\zeta} Y_t + \theta \beta \hat{A}_{t+1}^{p2}}$$

Approximation algorithm Once again, the modified model implies modifications to the solution algorithm. In the first place, the new aggregate price equation (3.25) is introduced, replacing the previous pricing rule (3.13). For approximating this new equation, the two additional neurons need to be calculated in every iteration of the Model simulation function. Steady-state values, too, need to be recalculated and replaced into the initialisation.

4 Simulations and results

Having described the base model and its extension, the projection method and its application, we now aim to show it in practice, simulating two shocks. We start with an impulse-response shock to productivity, analysing and comparing results between the different versions of the model. Secondly, an external demand shock is simulated, once again comparing the results between versions.

The simulation presumes the obtainment of the model's parameter values, usually done via calibration as in the RBC tradition, although more recent econometric methods of estimating dynamic stochastic general equilibrium models have been advanced. This work, however, relies on the already calibrated values of Lim and McNelis (2008). Additional details on their values can be found in Appendix (6.1).

4.1 Productivity shock

The productivity shock, or impulse-response, was implemented firstly by starting the simulation with Z in its stationary value, while setting productivity's disturbance parameter, ε^Z, equal to zero for all periods except for the moment of the shock, where it is given a positive value. Finally, the rest of the model was simulated, with the results shown in Figure 4.1.

The first result to notice is the direction of the effects being the same for both versions of the model, although with different magnitudes. The increase in productivity can be seen as having a corresponding increase in production, real wages and consumption, while both production factors' usage decrease and the price level falls. Simultaneously, the economy's trade balance and external financial position deteriorate, as the exchange rate appreciates. Finally, the central bank's Taylor rule causes an decrease in interest rates, in response to the drop in prices.

The size of the effects, however, varies between the two versions of the model. Firstly, the stickiness prevents prices from falling as much as in the flexible prices case, while at the same time returning faster to steady-state values. Secondly, both the exchange rate and external indebtedness suffer a slight increase, with the opposite effect on the trade balance. As expected, the deviation from fully flexible prices seems to impose an efficiency cost on the economy, as both product and consumption don't increase as much due to the imperfect adjustment of prices.

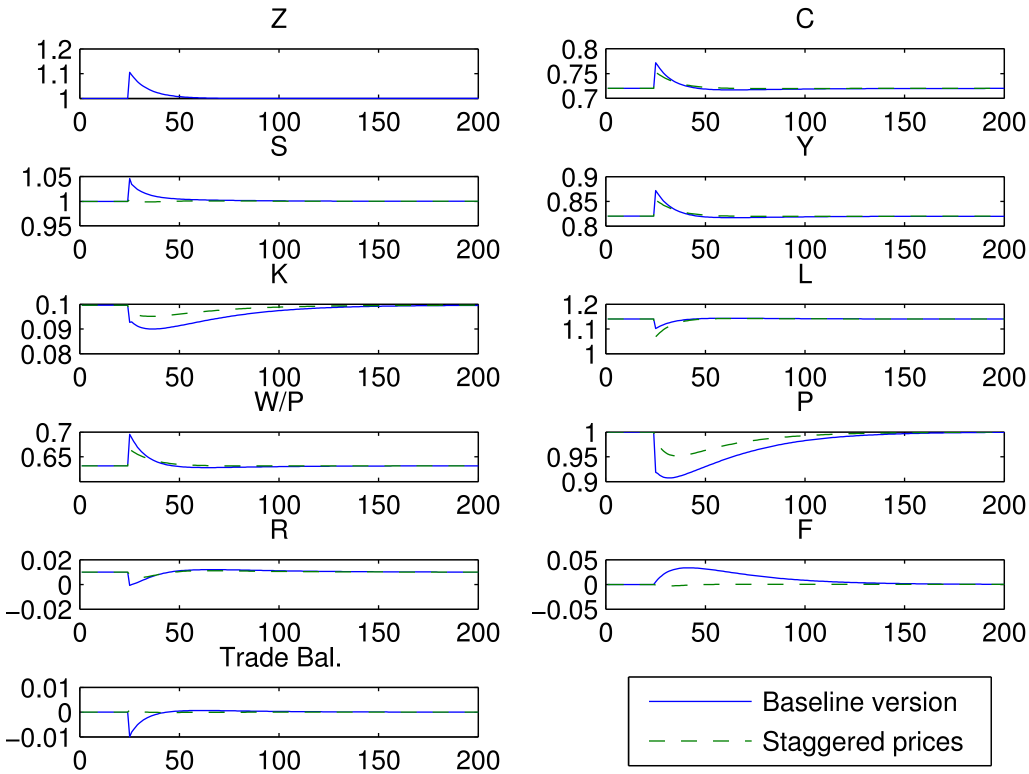

Figure 4.1: Productivity shock ($\varepsilon^Z = 0.1$)

4.2 External demand shock

For our second shock, we simulate the result of an increase of external demand, or an increase of exports. Being an exogenous constant variable, manipulating exports is a matter of setting X_t to the desired value. However, to add a measure of persistence in the shock, the constant variable is replaced with an autoregressive process tending to the steady-state value of exports:

$$\ln(X_t) = \gamma \ln(X_{t-1}) + (1-\gamma)\ln\left(\overline{X}\right) + \varepsilon_t^X \tag{4.1}$$

As in the productivity shock, the disturbance ε^X is set to the chosen value at the moment of the shock, while X_t is started from its stationary state value.

The results of the shock are shown in Figure 4.2.

First of all, the effect of an increase in external demand is an increase in aggregate de-

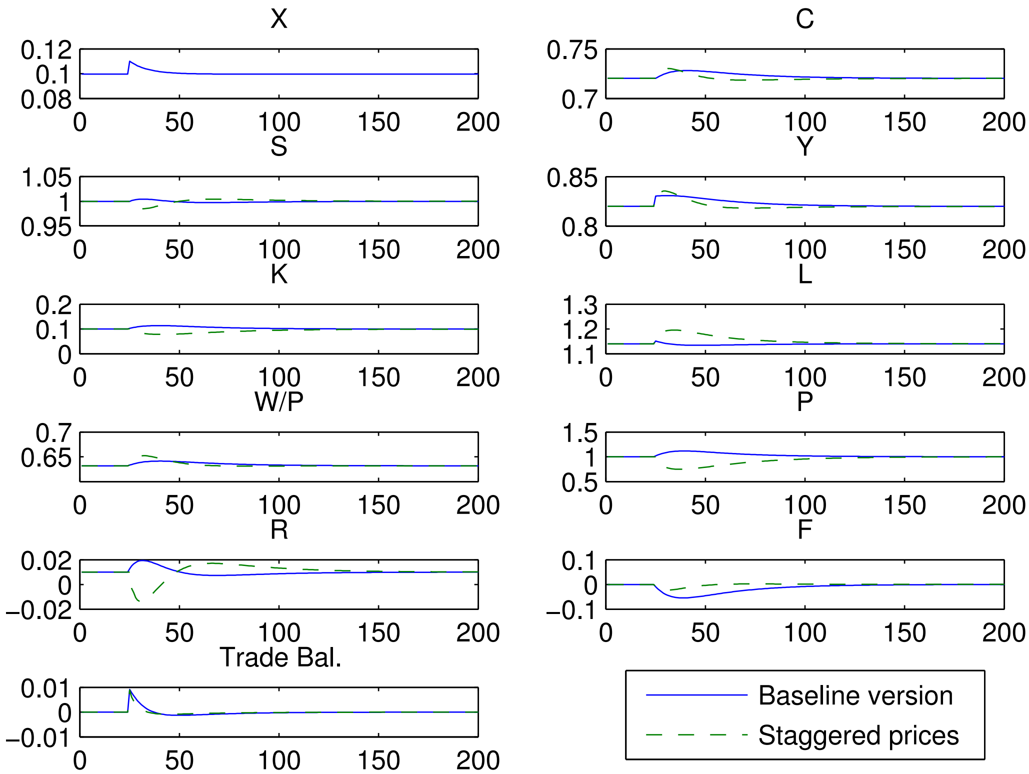

Figure 4.2: External demand increase ($\varepsilon^X = 0.1$)

mand, Y. In the flexible prices case, despite an increase in capital usage, which is fully imported, the trade balance also increases. Additionally, increases in labour usage, real wages and consumption can be seen in both versions of the model. As in the case of an productivity shock, the increase in product and consumption is lower with sticky prices than without, suggesting an efficiency loss from imperfect price adjustment.

Unlike the productivity shock case, however, the direction of several effects is different between the two versions of the model, most notably in the case of prices. While with flexible prices the increase in demand provokes a rise in prices, in the sticky prices version prices fall, with reflections in an higher increase of labour and wages, an decrease in capital usage, and exchange rate depreciation. Finally, in both versions the external demand shock results in an improvement of the external financial position, with households becoming creditors in respect to the rest of the world.

5 Conclusion

In the present work a DSGE model of the New Neoclassical Synthesis was derived in two versions: with and without sticky prices. For the solution of each version a projection method was employed, using an Artificial Neural Network as the approximation function, and minimising the sum of squared Euler errors for each problem. Finally, a set of impulse-response shocks were simulated on both already approximated versions of the model.

The correct solution of this model with the addition of imperfect competition and staggered prices is an important intermediate step for solving larger DSGE models, and for answering economic questions within their framework, given the prevalence of the referred extensions in most of the NNS literature. In this respect the projection method, although not widely used – mainly in favour of perturbation methods of different orders –, is shown to provide a good approximate solution, both as measured by the Euler errors and impulse-response results. In comparison to the said methods, the projection has the advantage of providing a global solution, in addition to fully retaining the nonlinearities of the approximated problem solution (depending on the applied approximation function).

The main difference between the algorithms used in each of this work's model versions, concerns the inclusion of an additional Euler equation to be approximated in the sticky prices version. This ease of extension exists because of the modularity of the original solution, which first draws the stochastic shocks and approximation function values, subsequently using MATLAB's nonlinear system solver *fsolve* for obtaining the rest of the variables, and finally calculating and iterating to minimise Euler errors. This allows any additional extensions to be further included simply by adding the corresponding equations, Euler errors and additional approximation functions to their respective places, without deep changes to the algorithm or modifications to the original description of the model.

As for future work, in the Economics front, a large literature on additional frictions and shocks exists, all trying to improve the fit to real data or the analysis of specific problems, such as the inclusion of money, consumption habits persistence, financial frictions, heterogeneous agents, and improved fiscal sections, for example, which could be explored with this method. In what concerns the application of the method itself, several speed improvements could be introduced, for example, by removing its nested loops through vectorisation of the MATLAB code. Finally, although operations within each model simulation are interrelated and not easily parallelised, iterations of the Monte Carlo method for drawing wealth distributions and variable correlations, a common practice in the literature, are fully separate from each other and can be done concurrently.

6 Appendices

6.1 Model parameters and steady-state

The model's parameters used in the simulations shown in this work are taken from Lim and McNelis (2008), and reproduced in the table below.

η	Relative risk aversion coefficient	1.5
ϖ	Elasticity of marginal disutility of labour	0.25
β	Discount factor	1/1.01
α	Production factors usage share	0.15
κ	Elasticity of substitution between production factors parameter	0.1
φ	Risk premium sensitivity	0.1
ρ	Productivity shock adjustment factor	0.9
ϕ_1	Taylor rule interest rate adjustment factor	0.9
ϕ_2	Taylor rule inflation sensitivity	1.5
ζ	Elasticity of substitution between differentiated consumption / final goods	6
θ	Staggered prices adjustment factor	0.85

Table 1: Model parameters

The steady-state of the model is calculated in a separate helper script for the parameters above, by removing indexes from all its variables, setting the productivity stochastic shock to a constant value, and solving the resulting system of equations.

6.2 MATLAB source code

In this section, selected parts from the MATLAB source code referred to in this work are reproduced, including the implementation described in the numerical solution subsections and used in the simulations of Chapter 4.

6.2.1 Baseline open economy model

```
% open_economy_model.m
%
% * function describing the open economy model, calculating C and S
%   based on an approximation
% ----------------------------------------------------------------

function [Error, C, F, K, L, P, R, S, W, Pk, Y, Z, tradebal, ...
    err_C, err_S, X] = open_economy_model(omega)

```

```
% global variables
global eta varpi beta1 alpha1 kappa rho phi2 phi1 varphi gamma1
global Rstar PF
global C_ss F_ss G_ss K_ss L_ss P_ss Pk_ss R_ss S_ss W_ss X_ss Y_ss Z_ss
global n_ini T1 T2 zshock n_state_vars n_neuron n_euler_eq xshock

% initialisations

% [multiple lines ommited here, mostly consisting of pre-allocating
%   the different arrays used next]

jk = n_state_vars*n_euler_eq*n_neuron;
jj = 1:n_state_vars:jk;
kk = n_state_vars:n_state_vars:jk;

% open economy system, to be solved with the fsolve function
% parameters:
%   x(1): Y;    x(2): L;    x(3): K;    x(4): Pk;
%   x(5): P;    x(6): W;    x(7): R;    x(8): F;
function y = model_system(x)
    y(1) = real((C(i,j)^-eta)*x(6) - (x(2)^varpi)*x(5));
    y(2) = real(Z(i,j)*((1 - alpha1)*(x(2)^kappa) + ...
        alpha1*(x(3)^kappa))^(1/kappa) - x(1));
    risk(i,j) = sign(x(8))*varphi*(exp(abs(x(8)) - F_ss) - 1);
    y(3) = S(i,j)*x(8) - (1 + Rstar + ...
        risk(i-1,j))*S(i,j)*F(i-1,j) - (S(i,j)*PF*x(3) - x(5)*X(i,j));
    y(4) = x(1) - C(i,j) - G_ss - X(i,j);
    y(5) = PF*S(i,j) - x(4);
    infl(i,j) = 0.25*((x(5)/P(i-1,j))^4 -1);
    y(6) = phi2*R(i-1,j) + (1 - phi2)*(Rstar + phi1*infl(i,j)) - x(7);
    y(7) = real(x(3) - ...
        x(2)*(x(6)*alpha1/((1 - alpha1)*x(4)))^(1/(1 - kappa)));
    aux1 = real(x(6)*((1 - alpha1) + ...
        alpha1*((x(6)*alpha1)/((1 - alpha1)*x(4)))^(kappa/(1 - ...
            kappa)))^(-1/kappa));
    aux2 = real(x(4)*(alpha1 + (1 - alpha1)*((x(4)*(1 - ...
        alpha1))/(alpha1*x(6)))^(kappa/(1-kappa)))^(-1/kappa));
    Mgc = 1/Z(i,j)*(aux1 + aux2);
    y(8) = x(5) - Mgc;
end

% model simulation
for j = 1:T2;
    for i = n_ini+1:T1
        % generating the productivity shock
        Zz = rho*log(Z(i-1,j))+(1-rho)*log(Z_ss)+zshock(i,j);
        Z(i,j) = exp(Zz);

        % exports adjustment function
        X(i,j) = gamma1*log(X(i-1,j)) + (1-gamma1)*log(X_ss) + xshock(i,j);
        X(i,j) = exp(X(i,j));

```

```
        % calculating the approximation function / neural network for
        %   obtaining C and S
        ZZ(i,j) = Z(i,j)-Z_ss;
        FF(i,j) = F(i-1,j)-F_ss;
        RR(i,j) = R(i-1,j)-R_ss;
        statevars = [ZZ(i,j) FF(i,j) RR(i,j)];

        for nn = 1: n_euler_eq*n_neuron;
            neuron(1,nn) = 1./(1+exp(-omega(jj(nn):kk(nn))*statevars'))-0.5;
        end;
        aux_C = neuron(1,1: n_neuron);
        aux_S = neuron(1,n_neuron+1:2*n_neuron);
        C(i,j) = exp(aux_C)*C_ss;
        S(i,j) = exp(aux_S)*S_ss;

        % solving the model using the C and S values given by the
        %   approximation
        s = fsolve(@model_system, [Y(i-1,j) L(i-1,j) K(i-1,j) Pk(i-1,j) P(i-1,j) ...
            W(i-1,j) R(i-1,j) F(i-1,j)]);
        Y(i,j) = s(1);
        L(i,j) = s(2);
        K(i,j) = s(3);
        Pk(i,j) = s(4);
        P(i,j) = s(5);
        W(i,j) = s(6);
        R(i,j) = s(7);
        F(i,j) = s(8);

        % calculating additional variables
        tradebal(i,j) = P(i,j)*X(i,j) - S(i,j)*Pk(i,j)*K(i,j);

        % Euler errors
        err_C(i,j) = real(C(i-1,j)^-eta)/P(i-1,j) - beta1*(real(C(i,j)^-eta)/P(i,j))*(1 + R(i-1,j));

        riskderiv(i,j) = sign(F(i,j))*varphi*(exp(abs(F(i,j))));
        if isinf(riskderiv(i,j))
            riskderiv(i,j) = exp(100);
        end
        err_S(i,j) = (1 + Rstar + risk(i-1,j) + riskderiv(i-1,j)*F(i-1,j))*S(i,j)/((1 + R(i-1,j))*S(i-1,j))
    end;
end;

err1 = reshape(err_C,T1*T2,1);
err2 = reshape(err_S,T1*T2,1);
Error = mean(err1.^2) + mean(err2.^2);
end
```

```
% open_economy_approx.m
%
% * calculates an approximation of the baseline model using the
%   projection method
% ---------------------------------------------------------------

clear all;

% name of the file where to load initial simulation parameters from
params_file = 'open_economy_params.csv';

% global variables
global eta varpi beta1 alpha1 kappa rho phi2 phi1 varphi gamma1
global Rstar PF
global C_ss F_ss G_ss K_ss L_ss P_ss Pk_ss R_ss S_ss W_ss X_ss Y_ss
global Z_ss n_ini T1 T2 zshock n_state_vars n_neuron n_euler_eq xshock

% model parameters
Rstar = 0.01;
PF = 1.0;
eta = 1.5;
varpi = 0.25;
beta1 = 1/1.01;
alpha1 = 0.15;
kappa = 0.1;
varphi = 0.1;
rho = 0.9;
phi2 = 0.9;
phi1 = 1.5;
gamma1 = 0.9;

% exogenous variables and steady-state values
G_ss = 0;
F_ss = 0;
P_ss = 1;
S_ss = 1;
Z_ss = 1;

Y_ss = 0.819853987226746;
L_ss = 1.140268178925569;
K_ss = 0.099605279053161;
Pk_ss = 1.000000000000000;
W_ss = 0.631648520484300;
R_ss = 0.010000000000000;
C_ss = 0.720248708173585;
X_ss = 0.099605279053161;

% number of periods and simulations
T1 = 400;
T2 = 2;
n_ini = 4;

```

```
% exports shock parameter
xshock = zeros(T1, T2);

% productivity stochastic shock
sd_shock = 0.01;
zshock = randn(T1,T2)*sd_shock;

% approximation function / neural network parameters
n_state_vars = 3;
n_euler_eq = 2;
n_neuron = 1;
nparm = n_state_vars*n_euler_eq*n_neuron;

% reading approximation function initial parameters
omega_params = dlmread(params_file);
omega_params = omega_params(end,:);

% approximation function optimisation
options = optimset('Display', 'iter', 'LargeScale', 'off');
omega_params = fminunc(@open_economy_model, omega_params, options);
disp(omega_params);
```

```
% open_economy_simul.m
%
% * describes the implementation of a set of possible impulse-response shocks
% ------------------------------------------------------------------------------

% [ initialisation ommited as it consists of the same done in the script above]

%% random productivity shock
sd_shock = 0.01;
zshock = randn(T1,T2)*sd_shock;

%% steady-state
zshock = zeros(T1, T2);

%% productivity impulse-response
zshock = [zeros(24,1); 0.1; zeros(T1-25,1)];

%% external demand impulse-response
xshock = [zeros(24,1); 0.1; zeros(T1-25,1)];

%% simulation and plot
[Error, C, F, K, L, P, R, S, W, Pk, Y, Z, tradebal, err_C, err_S] = ...
    open_economy_model(omega_params);
figure(1);
subplot(6,2,1);     plot(Z);      title('Z')
subplot(6,2,2);     plot(C);      title('C')
subplot(6,2,3);     plot(S);      title('S')
subplot(6,2,4);     plot(Y);      title('Y')
```

```
subplot(6,2,5);        plot(K);        title('K')
subplot(6,2,6);        plot(L);        title('L')
subplot(6,2,7);        plot(W./P);  title('W/P')
subplot(6,2,8);        plot(P);        title('P')
subplot(6,2,9);        plot(R);        title('R')
subplot(6,2,10);          plot(F);         title('F')
subplot(6,2,11);          plot(tradebal);  title('Trade Bal.');
```

6.2.2 Staggered prices version

```
% staggered_prices_model.m
%
% * function describing the staggered prices model, calculating C,
%   S and P based on an approximation
% ------------------------------------------------------------------------

function [Error, C, F, K, L, P, R, S, W, Pk, Y, Z, tradebal, A_p1, ...
    A_p2, err_C, err_S, err_A, X] = staggered_prices_model(omega)

% [ most of the initialisation ommited, except for additions or differences from open_economy_model.m ]

% global variables
global eta varpi beta1 alpha1 kappa rho phi2 phi1 varphi zeta theta
global Z_ss Ap1_ss Ap2_ss

% initialisations
A_p1 = Ap1_ss*ones(T1, T2);
A_p2 = Ap2_ss*ones(T1, T2);
err_A = zeros(T1, T2);

% staggered prices model system, to be solved with the fsolve function
% parameters:
%   x(1): Y
%   x(2): L
%   x(3): K
%   x(4): Pk
%   x(5): P
%   x(6): W
%   x(7): R
%   x(8): F
function y = model_system(x)
    y(1) = real((C(i,j)^-eta)*x(6) - (x(2)^varpi)*x(5));
    y(2) = real(Z(i,j)*((1 - alpha1)*(x(2)^kappa) + ...
        alpha1*(x(3)^kappa))^(1/kappa) - x(1));
    risk(i,j) = sign(x(8))*varphi*(exp(abs(x(8)) - F_ss) - 1);
    y(3) = S(i,j)*x(8) - (1 + Rstar + risk(i-1,j))*S(i,j)*F(i-1,j) - ...
        (S(i,j)*PF*x(3) - x(5)*X(i,j));
    y(4) = x(1) - C(i,j) - G_ss - X(i,j);
    y(5) = PF*S(i,j) - x(4);
    infl(i,j) = 0.25*((x(5)/P(i-1,j))^4 -1);
    y(6) = phi2*R(i-1,j) + (1 - phi2)*(Rstar + phi1*infl(i,j)) - x(7);
    y(7) = real(x(3) - x(2)*(x(6)*alpha1/((1-alpha1)*x(4)))^(1/(1-kappa)));
    Pnew = A_p1(i,j)/A_p2(i,j);
```

```
    y(8) = real(x(5) - (theta*P(i-1,j)^(1 - zeta) + (1 - theta)*Pnew^(1 - ...
        zeta))^(1/(1 - zeta)));
end

% model simulation
for j = 1:T2
    for i = n_ini:T1

        % [ same as in open_economy_model.m ]

        for nn = 1:n_euler_eq*n_neuron
            neuron(1,nn) = 1./(1+exp(-omega(jj(nn):kk(nn))*statevars'))-0.5;
        end
        aux_C = neuron(1, 1:n_neuron);
        aux_S = neuron(1, n_neuron+1 : 2*n_neuron);
        aux_Ap1 = neuron(1, n_neuron+2 : 3*n_neuron);
        aux_Ap2 = neuron(1, n_neuron+3 : 4*n_neuron);
        C(i,j) = exp(aux_C)*C_ss;
        S(i,j) = exp(aux_S)*S_ss;
        A_p1(i,j) = exp(aux_Ap1)*Ap1_ss;
        A_p2(i,j) = exp(aux_Ap2)*Ap2_ss;

        % solving the model using the C, S, Ap1 and Ap2 values given
        %   by the approximation
        s = fsolve(@model_system, ...
            [Y(i-1,j) L(i-1,j) K(i-1,j) Pk(i-1,j) P(i-1,j) W(i-1,j) R(i-1,j) F(i-1,j)]);

        % [ same as in open_economy_model.m ]

        % Euler errors
        err_C(i,j) = real(C(i-1,j)^-eta)/P(i-1,j) - ...
            beta1*(real(C(i,j)^-eta)/P(i,j))*(1 + R(i-1,j));

        riskderiv(i,j) = sign(F(i,j))*varphi*(exp(abs(F(i,j))));
        if isinf(riskderiv(i,j))
            riskderiv(i,j) = exp(100);
        end
        err_S(i,j) = (1 + Rstar + risk(i-1,j) + ...
            riskderiv(i-1,j)*F(i-1,j))*S(i,j)/((1 + R(i-1,j))*S(i-1,j)) - 1;

        aux_A1(i,j) = real((1 - alpha1) + alpha1*((W(i-1,j)*alpha1)/((1 - ...
            alpha1)*Pk(i-1,j)))^(kappa/(1 - kappa)));
        aux_A2(i,j) = real((1 - alpha1)*((Pk(i-1,j)*(1 - ...
            alpha1))/(alpha1*W(i-1,j)))^(kappa/(1 - kappa)) + alpha1);
        A(i,j) = real((1/Z(i-1,j))*(W(i-1,j)*aux_A1(i,j)^(-1/kappa) + ...
            Pk(i-1,j)*aux_A2(i,j)^(-1/kappa)));
        err_aux_Ap1(i,j) = real(Y(i-1,j)*A(i,j)*P(i-1,j)^zeta + beta1*theta*A_p1(i,j));
        err_aux_Ap2(i,j) = real(Y(i-1,j)*P(i-1,j)^zeta + beta1*theta*A_p2(i,j));
        err_A(i,j) = A_p1(i-1,j)/A_p2(i-1,j) - err_aux_Ap1(i,j)/err_aux_Ap2(i,j);
    end
end
```

References

Akerlof, G. A. and Yellen, J. L. (1985). A Near-Rational Model of the Business Cycle, with Wage and Price Inertia. *The Quarterly Journal of Economics*, 100(4):823–838.

Blanchard, O. J. (2008). neoclassical synthesis. In Durlauf, S. N. and Blume, L. E., editors, *The New Palgrave Dictionary of Economics*. Palgrave Macmillan, Basingstoke.

Calvo, G. A. (1983). Staggered prices in a utility-maximizing framework. *Journal of Monetary Economics*, 12(3):383–398.

Canova, F. (2007). *Methods for Applied Macroeconomic Research*. Princeton University Press.

Cass, D. (1965). Optimum Growth in an Aggregative Model of Capital Accumulation. *The Review of Economic Studies*, 32(3):233–240.

Chow, G. C. (1997). *Dynamic economics: optimization by the Lagrange method*. Oxford University Press, New York.

Clarida, R., Gali, J., and Gertler, M. (1999). The Science of Monetary Policy: A New Keynesian Perspective. *Journal of Economic Literature, American Economic Association*, 37(4):1661–1707.

Dixit, A. K. and Stiglitz, J. E. (1977). Monopolistic competition and optimum product diversity. *The American Economic Review*, 67(3):297–308.

Galí, J., López-Salido, J. D., and Vallés, J. (2007). Understanding the effects of government spending on consumption. *Journal of the European Economic Association*, 5(1):227–270.

Goodfriend, M. and King, R. G. (1997). The New Neoclassical Synthesis and the Role of Monetary Policy. In *NBER Chapters*, pages 231–296. National Bureau of Economic Research, Inc.

Heer, B. and Mauß ner, A. (2009). *Dynamic General Equilibrium Modeling*. Springer Berlin Heidelberg, Berlin, Heidelberg.

Hicks, J. (1937). Mr. Keynes and the "Classics"; A Suggested Interpretation. *Econometrica*, 5(2):pp. 147–159.

Judd, K. L. (1998). *Numerical Methods in Economics*. The MIT Press.

Kim, J. and Kim, S. H. (2003). Spurious welfare reversals in international business cycle models. *Journal of International Economics*, 60(2):471–500.

Koopmans, T. C. (1965). On the Concept of Optimal Economic Growth. In *The Economic Approach to Development Planning*. North-Holland, Amsterdam.

Kydland, F. E. and Prescott, E. C. (1982). Time to build and Aggregate fluctuations. *Econometrica*, 50(6):pp. 1345–1370.

Lim, G. C. and McNelis, P. D. (2008). *Computational Macroeconomics for the Open Economy*. The MIT Press.

Lucas, R. E. (1976). Econometric policy evaluation: A critique. *Carnegie-Rochester Conference Series on Public Policy*, 1:19–46.

Mankiw, N. G. (1985). Small Menu Costs and Large Business Cycles: A Macroeconomic Model of Monopoly. *The Quarterly Journal of Economics*, 100(2):529–537.

McCandless, G. T. (2008). *The ABCs of RBCs*. Harvard University Press.

Ramsey, F. P. (1928). A Mathematical Theory of Saving. *The Economic Journal*, 38(152):543–559.

Rebelo, S. (2005). Real business cycle models: past, present and future. *The Scandinavian Journal of Economics*, 107(2):217–238.

Schmitt-Grohé, S. and Uribe, M. (2004). Solving dynamic general equilibrium models using a second-order approximation to the policy function. *Journal of Economic Dynamics and Control*, 28(4):755–775.

Smets, F. and Wouters, R. (2003). An estimated dynamic stochastic general equilibrium model of the euro area. *Journal of the European Economic Association*.

Stokey, N. L. and Prescott, E. C. (1989). *Recursive Methods in Economic Dynamics*. Harvard University Press.

Swanson, E., Anderson, G., and Levin, A. (2005). Higher-Order Perturbation Solutions to Dynamic, Discrete-Time Rational Expectations Models: Methods and an Application to Optimal Monetary Policy. Computing in Economics and Finance 2005 146, Society for Computational Economics.

Taylor, J. B. (1979). Staggered Wage Setting in a Macro Model. *The American Economic Review*, 69(2):108–113.

Uhlig, H. F. (1995). *A toolkit for analyzing nonlinear dynamic stochastic models easily*. Institute for Empirical Macroeconomics, Federal Reserve Bank of Minneapolis.

Woodford, M. (2003). *Interest and prices: Foundations of a theory of monetary policy*. Princeton University Press.